

《操作系统真象还原》

图书基本信息

书名：《操作系统真象还原》

13位ISBN编号：9787115414343

出版时间：2016-3

作者：郑钢

页数：759

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

《操作系统真象还原》

内容概要

大学及研究生都有操作系统课程，这类人群具有很高的学术能力,但书中讲的过于抽象与晦涩，以至于很多学生对于此门课程恐惧到都提不出问题，只有会的人才能提出问题。操作系统理论书是无法让读者理解什么是操作系统的，学操作系统不能靠想像，他们需要看到具体的东西。

绝大多数技术人都对操作系统怀着好奇的心，他们渴望一本告诉操作系统到底是什么的书，里面不要掺杂太多无关的管理性的东西，代码量不大且是现代操作系统雏形，他们渴望很快看到本质而不花费大量的时间成本。

《操作系统真象还原》

作者简介

郑钢，毕业于北京大学，前百度运维高级工程师，对操作系统有深入的研究。好运动，喜钻研，热衷于尝试前沿技术，乐于分享学习成果。

书籍目录

第0章 一些你可能正感到迷惑的问题

1

0.1 操作系统是什么

1

0.2 你想研究到什么程度

2

0.3 写操作系统，哪些需要我来做

2

0.4 软件是如何访问硬件的

2

0.5 应用程序是什么，和操作系统是如何配合到一起的

3

0.6 为什么称为“陷入”内核

4

0.7 内存访问为什么要分段

4

0.8 代码中为什么分为代码段、数据段？这和内存访问机制中的段是一回事吗

6

0.9 物理地址、逻辑地址、有效地址、线性地址、虚拟地址的区别

11

0.10 什么是段重叠

12

0.11 什么是平坦模型

12

0.12 cs、ds这类sreg段寄存器，位宽是多少

12

0.13 什么是工程，什么是协议

13

0.14 为什么Linux系统下的应用程序不能在Windows系统下运行

14

0.15 局部变量和函数参数为什么要放在栈中

14

0.16 为什么说汇编语言比C语言快

15

0.17 先有的语言，还是先有的编译器，第1个编译器是怎么产生的

16

0.18 编译型程序与解释型程序的区别

19

0.19 什么是大端字节序、小端字节序

19

0.20 BIOS中断、DOS中断、Linux中断的区别	21
0.21 Section和Segment的区别	25
0.22 什么是魔数	29
0.23 操作系统是如何识别文件系统的	30
0.24 如何控制CPU的下一条指令	30
0.25 指令集、体系结构、微架构、编程语言	30
0.26 库函数是用户进程与内核的桥梁	33
0.27 转义字符与ASCII码	37
0.28 MBR、EBR、DBR和OBR各是什么	39
第1章 部署工作环境	42
1.1 工欲善其事，必先利其器	42
1.2 我们需要哪些编译器	42
1.2.1 世界顶级编译器GCC	42
1.2.2 汇编语言编译器新贵NASM	43
1.3 操作系统的宿主环境	43
1.3.1 什么是虚拟机	44
1.3.2 盗梦空间般的开发环境，虚拟机中再装一个虚拟机	45
1.3.3 virtualBox下载，安装	46
1.3.4 Linux发行版下载	46
1.3.5 Bochs下载安装	46
1.4 配置bochs	48
1.5 运行bochs	49
第2章 编写MBR主引导记录，让我们开始	

掌权	52
2.1 计算机的启动过程	52
2.2 软件接力第一棒，BIOS	52
2.2.1 实模式下的1MB内存布局	52
2.2.2 BIOS是如何苏醒的	54
2.2.3 为什么是0x7c00	56
2.3 让MBR先飞一会儿	58
2.3.1 神奇好用的\$和\$\$，令人迷惑的section	58
2.3.2 NASM简单用法	60
2.3.3 请下一位选手MBR同学做准备	60
第3章 完善MBR	65
3.1 地址、section、vstart浅尝辄止	65
3.1.1 什么是地址	65
3.1.2 什么是section	67
3.1.3 什么是vstart	68
3.2 CPU的实模式	70
3.2.1 CPU的工作原理	71
3.2.2 实模式下的寄存器	72
3.2.3 实模式下内存分段由来	76
3.2.4 实模式下CPU内存寻址方式	78
3.2.5 栈到底是什么玩意儿	81
3.2.6 实模式下的ret	84
3.2.7 实模式下的call	85
3.2.8 实模式下的jmp	

92	
3.2.9 标志寄存器flags	97
3.2.10 有条件转移	99
3.2.11 实模式小结	101
3.3 让我们直接对显示器说点什么吧	101
3.3.1 CPU如何与外设通信—IO接口	101
3.3.2 显卡概述	105
3.3.3 显存、显卡、显示器	106
3.3.4 改进MBR，直接操作显卡	110
3.4 bochs调试方法	112
3.4.1 bochs一般用法	113
3.4.2 bochs调试实例	118
3.5 硬盘介绍	122
3.5.1 硬盘发展简史	122
3.5.2 硬盘工作原理	123
3.5.3 硬盘控制器端口	126
3.5.4 常用的硬盘操作方法	128
3.6 让MBR使用硬盘	129
3.6.1 改造MBR	130
3.6.2 实现内核加载器	134
第4章 保护模式入门	136
4.1 保护模式概述	136
4.1.1 为什么要有保护模式	136
4.1.2 实模式不是32位CPU，变成了16位	137

4.2 初见保护模式	137
4.2.1 保护模式之寄存器扩展	137
4.2.2 保护模式之寻址扩展	140
4.2.3 保护模式之运行模式反转	141
4.2.4 保护模式之指令扩展	145
4.3 全局描述符表	150
4.3.1 段描述符	150
4.3.2 全局描述符表GDT、局部描述符表LDT及选择子	155
4.3.3 打开A20地址线	157
4.3.4 保护模式的开关，CR0寄存器的PE位	158
4.3.5 让我们进入保护模式	158
4.4 处理器微架构简介	165
4.4.1 流水线	166
4.4.2 乱序执行	168
4.4.3 缓存	168
4.4.4 分支预测	169
4.5 使用远跳转指令清空流水线，更新段描述符缓冲寄存器	172
4.6 保护模式之内存段的保护	173
4.6.1 向段寄存器加载选择子时的保护	173
4.6.2 代码段和数据段的保护	174
4.6.3 栈段的保护	175
第5章 保护模式进阶，向内核迈进	177
5.1 获取物理内存容量	

177	
5.1.1 学习Linux获取内存的方法	177
5.1.2 利用BIOS中断0x15子功能 0xe820获取内存	177
5.1.3 利用BIOS中断0x15子功能 0xe801获取内存	179
5.1.4 利用BIOS中断0x15子功能 0x88获取内存	180
5.1.5 实战内存容量检测	181
5.2 启用内存分页机制，畅游虚拟空间	186
5.2.1 内存为什么要分页	186
5.2.2 一级页表	188
5.2.3 二级页表	192
5.2.4 规划页表之操作系统与用户 进程的关系	197
5.2.5 启用分页机制	198
5.2.6 用虚拟地址访问页表	204
5.2.7 快表TLB (Translation Lookaside Buffer) 简介	206
5.3 加载内核	207
5.3.1 用C语言写内核	207
5.3.2 二进制程序的运行方法	211
5.3.3 elf格式的二进制文件	213
5.3.4 elf文件实例分析	218
5.3.5 将内核载入内存	222
5.4 特权级深入浅出	229
5.4.1 特权级那点事	229
5.4.2 TSS简介	

230	
5.4.3 CPL和DPL入门	
232	
5.4.4 门、调用门与RPL序	
235	
5.4.5 调用门的过程保护	
240	
5.4.6 RPL的前世今生	
243	
5.4.7 IO特权级	
248	
第6章 完善内核	
252	
6.1 函数调用约定简介	
252	
6.2 汇编语言和C语言混合编程	
256	
6.2.1 浅析C库函数与系统调用	
256	
6.2.2 汇编语言和C语言共同协作	
259	
6.3 实现自己的打印函数	
261	
6.3.1 显卡的端口控制	
261	
6.3.2 实现单个字符打印	
265	
6.3.3 实现字符串打印	
275	
6.3.4 实现整数打印	
277	
6.4 内联汇编	
281	
6.4.1 什么是内联汇编	
281	
6.4.2 汇编语言AT&T语法简介	
281	
6.4.3 基本内联汇编	
283	
6.4.4 扩展内联汇编	
284	
6.4.5 扩展内联汇编之机器模式简介	
294	
第7章 中断	
298	
7.1 中断是什么，为什么要有中断	
298	
7.2 操作系统是中断驱动的	
299	

7.3 中断分类	299
7.3.1 外部中断	299
7.3.2 内部中断	301
7.4 中断描述符表	304
7.4.1 中断处理过程及保护	306
7.4.2 中断发生时的压栈	308
7.4.3 中断错误码	310
7.5 可编程中断控制器8259A	311
7.5.1 8259A介绍	311
7.5.2 8259A的编程	314
7.6 编写中断处理程序	319
7.6.1 从最简单的中断处理程序开始	319
7.6.2 改进中断处理程序	335
7.6.3 调试实战：处理器进入中断时压栈出栈完整过程	339
7.7 可编程计数器/定时器8253简介	346
7.7.1 时钟—给设备打拍子	346
7.7.2 8253入门	348
7.7.3 8253控制字	349
7.7.4 8253工作方式	350
7.7.5 8253初始化步骤	353
7.8 提高时钟中断的频率，让中断来得更猛烈一些	354
第8章 内存管理系统	357
8.1 makefile简介	357

8.1.1 makefile是什么	357
8.1.2 makefile基本语法	358
8.1.3 跳到目标处执行	360
8.1.4 伪目标	361
8.1.5 make：递归式推导目标	362
8.1.6 自定义变量与系统变量	363
8.1.7 隐含规则	365
8.1.8 自动化变量	366
8.1.9 模式规则	367
8.2 实现assert断言	367
8.2.1 实现开、关中断的函数	367
8.2.2 实现ASSERT	370
8.2.3 通过makefile来编译	372
8.3 实现字符串操作函数	374
8.4 位图bitmap及其函数的实现	377
8.4.1 位图简介	377
8.4.2 位图的定义与实现	378
8.5 内存管理系统	381
8.5.1 内存池规划	381
8.5.2 内存管理系统第一步，分配页内存	388
第9章 线程	398
9.1 实现内核线程	398
9.1.1 执行流	398
9.1.2 线程到底是什么	399

9.1.3 进程与线程的关系、区别简述	402
9.1.4 进程、线程的状态	405
9.1.5 进程的身份证—PCB	405
9.1.6 实现线程的两种方式—内核或用户进程	406
9.2 在内核空间实现线程	409
9.2.1 简单的PCB及线程栈的实现	409
9.2.2 线程的实现	413
9.3 核心数据结构，双向链表	417
9.4 多线程调度	421
9.4.1 简单优先级调度的基础	421
9.4.2 任务调度器和任务切换	425
第10章 输入输出系统	439
10.1 同步机制——锁	439
10.1.1 排查GP异常，理解原子操作	439
10.1.2 找出代码中的临界区、互斥、竞争条件	444
10.1.3 信号量	445
10.1.4 线程的阻塞与唤醒	447
10.1.5 锁的实现	449
10.2 用锁实现终端输出	452
10.3 从键盘获取输入	456
10.3.1 键盘输入原理简介	456
10.3.2 键盘扫描码	457
10.3.3 8042简介	463
10.3.4 测试键盘中断处理程序	

465	
10.4	编写键盘驱动
468	
10.4.1	转义字符介绍
468	
10.4.2	处理扫描码
469	
10.5	环形输入缓冲区
476	
10.5.1	生产者与消费者问题简述
476	
10.5.2	环形缓冲区的实现
478	
10.5.3	添加键盘输入缓冲区
481	
10.5.4	生产者与消费者实例测试
482	
第11章	用户进程
485	
11.1	为什么要有任务状态段TSS
485	
11.1.1	多任务的起源，很久很久以前.....
485	
11.1.2	LDT简介
486	
11.1.3	TSS的作用
488	
11.1.4	CPU原生支持的任务切换方式
492	
11.1.5	现代操作系统采用的任务切换方式
495	
11.2	定义并初始化TSS
497	
11.3	实现用户进程
501	
11.3.1	实现用户进程的原理
501	
11.3.2	用户进程的虚拟地址空间
501	
11.3.3	为进程创建页表和3特权级栈
502	
11.3.4	进入特权级3
505	
11.3.5	用户进程创建的流程
506	
11.3.6	实现用户进程—上

507	
11.3.7	bss简介
513	
11.3.8	实现用户进程一下
515	
11.3.9	让进程跑起来—用户进程的调度
519	
11.3.10	测试用户进程
520	
第12章	进一步完善内核
523	
12.1	Linux系统调用浅析
523	
12.2	系统调用的实现
527	
12.2.1	系统调用实现框架
527	
12.2.2	增加0x80号中断描述符
527	
12.2.3	实现系统调用接口
528	
12.2.4	增加0x80号中断处理例程
528	
12.2.5	初始化系统调用和实现 sys_getpid
530	
12.2.6	添加系统调用getpid
531	
12.2.7	在用户进程中的系统调用
532	
12.2.8	系统调用之栈传递参数
534	
12.3	让用户进程“说话”
536	
12.3.1	可变参数的原理
536	
12.3.2	实现系统调用write
538	
12.3.3	实现printf
539	
12.3.4	完善printf
542	
12.4	完善堆内存管理
545	
12.4.1	malloc底层原理
545	
12.4.2	底层初始化
548	

12.4.3 实现sys_malloc	550
12.4.4 内存的释放	555
12.4.5 实现sys_free	558
12.4.6 实现系统调用malloc和free	562
第13章 编写硬盘驱动程序	566
13.1 硬盘及分区表	566
13.1.1 创建从盘及获取安装的 磁盘数	566
13.1.2 创建磁盘分区表	567
13.1.3 磁盘分区表浅析	571
13.2 编写硬盘驱动程序	578
13.2.1 硬盘初始化	578
13.2.2 实现thread_yield和idle线程	582
13.2.3 实现简单的休眠函数	584
13.2.4 完善硬盘驱动程序	585
13.2.5 获取硬盘信息，扫描分区表	590
第14章 文件系统	595
14.1 文件系统概念简介	595
14.1.1 inode、间接块索引表、文件 控制块FCB简介	595
14.1.2 目录项与目录简介	597
14.1.3 超级块与文件系统布局	599
14.2 创建文件系统	601
14.2.1 创建超级块、i结点、目录项	601
14.2.2 创建文件系统	603
14.2.3 挂载分区	

609	
14.3 文件描述符简介	612
14.3.1 文件描述符原理	612
14.3.2 文件描述符的实现	614
14.4 文件操作相关的基础函数	615
14.4.1 inode操作有关的函数	616
14.4.2 文件相关的函数	620
14.4.3 目录相关的函数	623
14.4.4 路径解析相关的函数	628
14.4.5 实现文件检索功能	630
14.5 创建文件	633
14.5.1 实现file_create	633
14.5.2 实现sys_open	636
14.5.3 在文件系统上创建第1个文件	639
14.6 文件的打开与关闭	640
14.6.1 文件的打开	640
14.6.2 文件的关闭	642
14.7 实现文件写入	643
14.7.1 实现file_write	643
14.7.2 改进sys_write及write系统调用	648
14.7.3 把数据写入文件	650
14.8 读取文件	651
14.8.1 实现file_read	651
14.8.2 实现sys_read与功能验证	653

14.9 实现文件读写指针定位功能	655
14.10 实现文件删除功能	657
14.10.1 回收inode	657
14.10.2 删除目录项	660
14.10.3 实现sys_unlink与功能验证	663
14.11 创建目录	665
14.11.1 实现sys_mkdir创建目录	666
14.11.2 创建目录功能验证	669
14.12 遍历目录	671
14.12.1 打开目录和关闭目录	671
14.12.2 读取1个目录项	673
14.12.3 实现sys_readdir及sys_rewinddir	674
14.13 删除目录	676
14.13.1 删除目录与判断空目录	676
14.13.2 实现sys_rmdir及功能验证	677
14.14 任务的工作目录	679
14.14.1 显示当前工作目录的原理及基础代码	679
14.14.2 实现sys_getcwd	681
14.14.3 实现sys_chdir改变工作目录	683
14.15 获得文件属性	684
14.15.1 ls命令的幕后功臣	684
14.15.2 实现sys_stat	685
第15章 系统交互	687
15.1 fork的原理与实现	

687	
15.1.1	什么是fork
687	
15.1.2	fork的实现
689	
15.1.3	添加fork系统调用与实现init 进程
695	
15.2	添加read系统调用，获取键盘输入
696	
15.3	添加putchar、clear系统调用
697	
15.4	实现一个简单的shell
699	
15.4.1	shell雏形
699	
15.4.2	添加Ctrl+u和Ctrl+l快捷键
701	
15.4.3	解析键入的字符
703	
15.4.4	添加系统调用
705	
15.4.5	路径解析转换
708	
15.4.6	实现ls、cd、mkdir、ps、rm等 命令
712	
15.5	加载用户进程
717	
15.5.1	实现exec
717	
15.5.2	让shell支持外部命令
723	
15.5.3	加载硬盘上的用户程序执行
724	
15.5.4	使用户进程支持参数
727	
15.6	实现系统调用wait和exit
731	
15.6.1	wait和exit的作用
731	
15.6.2	孤儿进程和僵尸进程
732	
15.6.3	一些基础代码
733	
15.6.4	实现wait和exit
737	
15.6.5	实现cat命令
741	

15.7 管道

745

15.7.1 管道的原理

745

15.7.2 管道的设计

747

15.7.3 管道的实现

748

15.7.4 利用管道实现进程间通信

752

15.7.5 在shell中支持管道

754

参考文献

760

《操作系统真象还原》

精彩短评

- 1、入门简单
- 2、挺好 通俗易懂
- 3、对我这种基础不好的人来说，很详细，很多东西解释清楚了。有时候你还要知道为什么这么做，这个很重要
- 4、十分详细,但有些内容作者并没有讲的很详细,比如处理器的体系结构,建议配合CSAPP阅读更佳
- 5、讲得很清楚,但要有一定的背景知识才能看懂,书的知识点有点散(可能是叙述风格造成的),如果书前言能提到如何阅读本书和面向哪些读者那就更好了

《操作系统真象还原》

精彩书评

1、大二下学期开设了操作系统课程，课上理论过多，缺乏实践，对操作系统的运行机制依旧茫然。于是在知乎看到如何自己写一简单内核，了解到需要汇编。自己买了王爽汇编自学，然而那本汇编讲的只是16位的8086汇编，在别人推荐下，买了一本x86从实模式到保护模式，读完之后会对实模式和保护模式、系统的基本引导有基本的了解，后来在china-pub上再看《《操作系统真象还原》>>看了目录。感觉内容很不错，拿到手后确实很厚，不过不用担心，之所以这么厚。作者用非常通俗，幽默得讲解每一个概念，知识点。而且不断提及，生怕读者弄不懂，读起来很舒服，作者真是煞费苦心，为作者点赞，这本书有一个非常好的好处就是。每章对应的代码实现一新的功能，可以用bochs调试，自己照着书上抄有时也会出现错误，需要你自己动手调试，对比作者源码，这样印象会很深刻。这样做之前在其那些很晦涩的概念，自己阅读，调试后，感觉也就那么回事。随着深入对操作系统的全貌都有了清晰的认识，深刻体会到了内核实现的各种巧妙思想。看得我直拍大腿。这本书也不是完全面向0基础,需要有了一点汇编/c语言,跟基本的数据结构(链表),如果不会汇编.推荐王爽的汇编语言跟x86从实模式到保护模式,再看这本书会比较快.另外此书有一点瑕疵就是elf格式那块讲得不是很明白.推荐coursera南京大学开设的程序的表示、转换与链接(<https://www.coursera.org/learn/jisuanji-xitong>),跟深入理解计算机系统>的第7章链接知识.

《操作系统真象还原》

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:www.tushu000.com